CLAIMS

What is claimed is:

1. A method for optimizing the use of a plurality of processors when compiling a program in a computer system, the method comprising the steps of:

(a) providing a list of directories and a list of processors;

(b) assigning a directory to a next available processor in an ordered manner to allow the next available processor to compile at least one file within the directory; and

(c) repeating step (b) to ensure that the maximum number of directories can be compiled.

2. The method of claim 1 wherein the assigning step (b) further includes the step of (b1) obtaining a directory in which all dependencies have been satisfied.

3. The method of claim 1 wherein the assigning step (b) further includes the step of (b1), updating the list of processors and the list of directories based upon the assignment of the directory.

4. The method of claim 1 wherein the assigning step (b) further includes the step of (b1) providing a directory update mechanism for assigning the directories in the ordered manner.

5. The method of claim 4 wherein the providing an update mechanism step (b1)

further comprises the steps of:

(b11) providing an array of dependency changes; and

(b12) merging the dependency changes array with a master array of changes.

6. The method of claim 5 wherein the merging step (b12) comprises the steps of:

(b121) obtaining a dependency change from the dependency changes array;

(b122) determining whether the dependency change is in a directory in the master array;

(b123) updating the directory in the master array of the dependency change in a directory of the master array;

(b124) adding dependency change to the master array in a new directory if the dependency change is not in a directory of the master array;

(b125) determining if there is another dependency change in the dependency changes array after either step (b123) or step (b124); and

(b126) repeating steps (b121) – (b125) until all dependency changes have been obtained from the dependency change array.

7. A system for optimizing the use of a plurality of processors when compiling a program in a computer system, the system comprising:

means for providing a list of directories and a list of processors;

means for assigning a directory to a next available processor in an ordered manner to allow the next available processor to compile at least one file within the directory; and

means for ensuring that the maximum number of directories can be compiled.

8. The system of claim 7 wherein the assigning means further includes means for obtaining a directory in which all dependencies and prerequisites have been satisfied.

9. The system of claim 7 wherein the assigning means further includes the means for updating the list of processors and the list of directories based upon the assignment of the directory.

10. The system of claim 10 wherein the assigning means further includes the means for providing a directory update mechanism for assigning the directories in the ordered manner.

11. The system of claim 10 wherein the providing an update mechanism means further comprises:

means for providing an array of dependency changes; and

means for merging the dependency changes array with a master array of changes.

12. The method of claim 5 wherein the merging means comprises:

means for obtaining a dependency change from the dependency changes array;

means for determining whether the dependency change is in a directory in the master array;

means for updating the directory in the master array of the dependency change in a directory of the master array;

means for adding dependency change to the master array in a new directory if the dependency change is not in a directory of the master array;

means for determining if there is another dependency change in the dependency changes array; and

means for ensuring that all dependency changes have been obtained from the dependency change array.

13. A computer readable medium containing program instructions for optimizing the use of a plurality of processors when compiling a program in a computer system, the program instructions for:

(a) providing a list of directories and a list of processors;

(b) assigning a directory to a next available processor in an ordered manner to allow the next available processor to compile at least one file within the directory; and

(c) repeating step (b) to ensure that the maximum number of directories can be compiled.

14. The computer readable medium of claim 13 wherein the assigning step (b) further includes the step of (b1) obtaining a directory in which all dependencies and prerequisites have been satisfied.

15. The computer readable medium of claim 131 wherein the assigning step (b) further includes the step of (b1), updating the list of processors and the list of directories based upon the assignment of the directory.

16. The computer readable medium of claim 13 wherein the assigning step (b)

further includes the step of (b1) providing a directory update mechanism for assigning the directories in the ordered manner.

17. The computer readable medium of claim 16 wherein the providing an update mechanism step (b1) further comprises the steps of:

(b11) providing an array of dependency changes; and

(b12) merging the dependency changes array with a master array of changes.

18. The computer readable medium of claim 17 wherein the merging step (b12) comprises the steps of:

(b121) obtaining a dependency change from the dependency changes array;

(b122) determining whether the dependency change is in a directory in the master array;

(b123) updating the directory in the master array of the dependency change in a directory of the master array;

(b124) adding dependency change to the master array in a new directory if the dependency change is not in a directory of the master array;

(b125) determining if there is another dependency change in the dependency changes array after either step (b123) or step (b124); and

(b126) repeating steps (b121) – (b125) until all dependency changes have been obtained from the dependency change array.